VB-Server zur Ankopplung an Cross

Ab VKR C1, R1.1.8 werden „offline“ Programme mit dem ooP Server CrossCommExe.exe an das Grundsystem angekoppelt. Der Server bietet sich auch zur einfachen Programmierung von production screens an.

1. Allgemeine Einstellungen

1.1 TimeOut

- Vorgabe in [ms]
- falls kein Wert uebergeben wird, wird Standard timeout verwendet
- Standardtimeout: Directory 10000, Up- Download 70000 alle anderen 3000ms

1.2 Server

- Server ist ceateable single use
- Registrieren durch Start Server

1.3 Fehlerhandling

- falls Kommunikatio niO, liefern die Funktionen FALSE zurueck
- nach Fehler ist neuer connect erforderlich

2. Schnittstelle:

2.1. Initialisierung

Dim CrossCommands As Object
Set CrossCommands = CreateObject("CrossCommEXE.CrossCommand")
CrossCommands.Init Me

2.2. Verbindungsaufbau

Public Function ConnectToCross(ByVal sConnectName As String, Optional nC_Mode as integer) As Boolean
Bsp.:
strResult = CrossCommands.ConnectToCross("example")
Hinweis: nC_Mode ist als Default = 0 (Cross-Anmeldung ohne aynchrone Meldungen, -1 = mit async. Meld.)

2.3. Verbindungsende

Public Sub ServerOff()
Bsp.:
CrossCommands.ServerOff()
Hinweise:
- der Server wird beendet
- alle offenen SetInfos werden beendet

2.4. Filehandling

2.4.1
Public Function GetRobotDirectory(ByRef sDir As String, Optional vMask, Optional vTimeOut)
Bsp.:
strResult = CrossCommands.GetRobotDirectory(strParam, "folge*.src")
Hinweise:
- grundsaetzlich nur /R1 moeglich
- strParam enthaelt alle Dateinamen, durch CHR(10) getrennt

2.4.2
Public Function DownLoadDiskToRobot(ByVal sFileName As String, Optional vTimeOut)
Bsp.:

strResult = CrossCommands.DownLoadDiskToRobot("c:\programme\krc\poweron\r1\cell.src", 5000)
Hinweise:
- grundsaetzlich nur /R1 moeglich

2.4.3
Public Function DownLoadMemToRobot(ByVal sFileName As String, ByVal sFile As String, Optional vTimeOut)
Bsp.:
strResult = CrossCommands.DownLoadMemToRobot("cell.src", strFile)
Hinweise:
- grundsaetzlich nur /R1 moeglich
- strFile enthaelt das File als string

2.4.4
Public Function UpLoadFromRobotToDisk(ByVal sFileName As String, ByVal sPath As String, _
Optional sOptions, Optional vTimeOut) As Boolean
Bsp.:
strResult = CrossCommands.UpLoadFromRobotToDisk("cell.src", "c:\", 5000)
Hinweise:
- sOptions kann „-s“ (Systemdateien enthalten), Standard ohne

2.4.5
Public Function UpLoadFromRobotToMem(ByVal sFileName As String, ByRef sFile As String, _
Optional sOptions, Optional vTimeOut) As Boolean
Bsp.:
strResult = CrossCommands.UpLoadFromRobotToMem("cell.src", strFile, "-s", 5000)
Hinweise:
- sOptions kann „-s“ (Systemdateien enthalten) , Standard ohne

2.4.6
Public Function DeleteRobotProgram(ByVal sPrgName As String, Optional vTimeOut) As Boolean
Bsp.:
strResult = CrossCommands.DeleteRobotProgram("cell.src")
Hinweise:
- grundsaetzlich nur /R1 moeglich
- keine Nachfrage

2.5 Variablenhandling

2.5.1
Public Function ShowVar(ByVal sVariableName As String, ByRef sResult As String, Optional vTimeOut) As Boolean
Bsp.:
strResult = CrossCommands.ShowVar("$PRO_STATE0", strParam)

2.5.2
Public Function SetVar(ByVal sVariableName As String, ByVal sNewValue As String, Optional vTimeOut) As Boolean
Bsp.:
strResult = CrossCommands.SetVar("$mode_op", "#T1")

2.5.3
Public Function SetInfoOn(ByVal sVariableName As String, ByRef sResult As String, Optional vTimeOut) As Boolean
Bsp.:
strResult = CrossCommands.SetInfoOn("$mode_op", strparam)
Hinweise:
- max. 4 SetInfos gleichzeitig
- SetInfos auf gleiche Variablen werden ignoriert

2.5.4

Public Function SetInfoOff(ByVal sVariableName As String, Optional vTimeOut) As Boolean
Bsp.:
strResult = CrossCommands.SetInfoOff("$mode_op")

2.6. Programmhandling

2.6.1
Public Function RobotLevelStop(Optional vTimeOut) As Boolean
Bsp.:
strResult = CrossCommands.RobotLevelStop()

2.6.2
Public Function ControlLevelStop(Optional vTimeOut) As Boolean
Bsp.:
strResult = CrossCommands.ControlLevelStop()

2.6.3
Public Function RunControlLevel(Optional vPrgName, Optional vTimeOut) As Boolean
Bsp.:
strResult = CrossCommands.RunControlLevel()
Hinweise:
- falls vPrgName fehlt, wird Name aus $machine.dat benutzt

2.6.4
Public Function SelectModul(ByVal strFile, Optional ByVal strParam, Optional bImplizitCancel,
Optional vTimeOut) As Boolean
Bsp.:
strResult = CrossCommands.SelectModul („HUGO.SRC“, ,True)
Hinweis:
- strParam sind die Übergabeparameter des aufgerufenen Modules
- bImplicitCancel ist als Default auf False gesetzt.

2.6.5

Public Function CancelModul(Optional vTimeOut) As Boolean
Bsp.:
strResult = CrossCommands.CancelModul ()
Hinweis:
- es wird das aktuell angewählte Programm im R_INT abgewählt

2.7. Systemhandling

2.7.1
Public Function ConfirmAll(Optional vTimeOut) As Boolean
Bsp.:
strResult=ConfirmAll(3000)

2.7.2
Public Function KrcOk(Optional vTimeOut) As Boolean
Bsp.:
strResult=KrcOk(3000)

2.7.3
Public Function IO_Restart(ByVal nBus As IO_TYPE, Optional vTimeOut) As Boolean
Bsp.:
strResult = CrossCommands.IO_Restart (-1)
Hinweis: die Übergabenummer entspricht den folgenden Konstanten
IORestart = -1
IBusReset = 1
CanReset = 2
BoschReset = 3
PerceptronReset = 4

2.8 Properties

2.8.1
Public Property Get CrossIsConnected() As Boolean
Hinweise:
- gibt nur aktuellen Zustand wieder, keine Pruefung

2.8.2
Public Property Get CrossError() As Long
Hinweise:
- letzter cross error

2.9 Client Routinen

2.9.1
Public Sub SetInfoResult(ByRef strResult As String)
End Sub
Hinweise:
- Verarbeitung der SetInfo Ergebnisse im Client

2.9.2
Public sub BOFKey (byref nKey as integer)
Hinweis:
- Verarbeitung der KCP - Softkey´s, Statuskey´s (links)

2.9.3
RobMessage (byref eMsg Msg_Type, byref strMeldung as string, byref pnVerwaltung as string)
- Asynchrone Meldungen fall beim ConnectToCross mit nC_Mode=-1 angemeldet wurden.